# Information Processing Techniques
# FORTRAN-lint[1] (Fortran-77 Source Code Analyzer) Evaluation

*Terance L. Lam*
*Computer Sciences Corporation*
*Numerical Aerodynamic Simulation Division*
*NASA Ames Research Center*

*January 14, 1992*

*Abstract*

*This document discusses the evaluation of Information Processing Techniques' FORTRAN-lint (Fortran-77 Source Code Analyzer) on the SGI 4D workstation. Special features of Flint are discussed and a cost analysis is performed. It is concluded that the advantages of FORTRAN-lint are outweighed by the advantages of NAGWare F77 Tools; therefore, it is recommended that FORTRAN-lint not be purchased.*

## 1.0 Introduction

FORTRAN-lint (Flint) is a Fortran-77 (ANSI X3.9 1978) source code analyzer developed by Information Processing Techniques. This source code analyzer is similar to lint, the C program checker. It is used to examine source files associated with a Fortran-77 program and to report program bugs before the program is compiled. FORTRAN-lint checks for the following five general classes of problems:

- syntax problems (e.g., a re-dimensioned array)
- call-interface problems—data type usage conflicts across different modules and problems related to subroutine or function calls and/or argument passing
- data-usage problems—problems such as variables or common block members

1. FORTRAN-lint is a registered trademark of Information Processing Techniques, Inc.

used but not set and inconsistencies in common-block declarations

- source-code-portability restrictions involved in moving Fortran programs between different machines and/or operating systems (e.g., a jump into a DO loop)
- problems related to implicitly typed variables, unused functions, subroutines, and variables

This evaluation examines how Flint might be a useful tool for NAS users. Also, some of the major features of Flint will be addressed.

## 2.0 Flint

Flint follows the UNIX command syntax convention; therefore, there is no need for users to learn a new command language. This feature highly enhances ease of use. The software package comes with a 40 page manual and on-line manual pages that make references easy.

Flint classifies syntax problems, call-interface problems, and data-usage problems into three levels of severity and reports them accordingly:

- hint, or "for your information" (FYI), messages
- warning messages
- error messages.

### 2.1 Installation

Installation of the FORTRAN-lint involves:

- loading the software from the tape
- executing the installation shell script "install_flint"
- setting up an environment variable to point to the installation directory
- calling up IPT to obtain an CPU specific authorization code to activate Flint.

Once these procedures are executed, the installation is complete and Flint is ready for use. Installing Flint was considered to be easy, especially with the installation shell script. An installation only took about five minutes and succeeded at the first trial.

## 2.2 Evaluation

Since this evaluation software was available only for two weeks, it was difficult to perform a complete evaluation by going through a normal Fortran-code-development cycle in such a short period of time. The best that could be done was applying this analyzer to some of the in-production software to explore functionality and several of its major features.

Three Fortran-77 application programs were analyzed: Eagle-Grid, Eagle-Surf, and Ingrid. Each of these applications consisted of more than a thousand lines of Fortran code, organized into sub-modules.

The analysis is invoked with the following command:

```
% flint -agfstx -S -E files.lst
```

The command exercises some of the more useful options of Flint, shown below:

- -E Flint accepts one or more Fortran-77 source files, each of which may contain one or more Fortran-77 sub-programs. When Flint is invoked with this option, it analyzes all files included in the input file.
- -a Portability Check. It checks for program's conformance to the Fortran-77 ANSI standard.
- -g Performs global check on a group of related modules for call interface, common block usage and local variable usage conflicts; otherwise, Flint analyzes (local mode) programs as individual modules.
- -f Generates extra hints during source analysis.
- -t Performs flow analysis to generate a call tree of the calling structure.
- -x Generates cross reference table and symbol table.
- -s Summarizes analysis statistics.
- -S Flint displays result of the analysis on the stdout. With the "-S" switch, Flint will redirect output from stdout to the following files:

  Name.LNTAnalysis output
  Name.XRFXreference table (-x)
  Name.TRECall tree (-t)
  Name.SUMSummary(-+)

Name.STTstatistics(-s)

## 2.3 Performance

Flint's performance is efficient. Its execution does not reduce workstation performance. Table 1 summarizes of the results of the analysis of these packages.

| | Eagle Grid | Eagle Surf | Ingrid |
|---|---|---|---|
| Number of Source files | 121 | 65 | 56 |
| Total Number of Lines | 24,087 | 15,549 | 7,871 |
| Number of Syntax messages | 0 | 4 | 0 |
| Number of Interface messages | 3 | 2 | 75 |
| Number of Data Usage messages | 4 | 2 | 6 |
| Number of ANSI-F77 Port Problems | 138 | 671 | 149 |
| CPU Time (user time in seconds) | 14.6 | 8.6 | 6.7 |

Table 1.Summary of Analysis of Eagle-Grid, Eagle-Surf, and Ingrid

The output files of the analysis were examined. The majority of the discrepancies reported involved the interface, such as problems related to subroutine or function calls and/or argument passing, variables or common block members used but not set, etc.

In general, Flint reports its errors and informational messages in a more understandable manner than the conventional F77 compiler, as shown in Figures 1, 2 and 3. Figure 1 is a listing of part of the program; Figures 2 and 3 are error messages reported by the SGI 4D F77 compiler and Flint respectively.

```
      Line 68     PIKS+=200.

      Line 69     VDGS=500;

      Line 70fdsf RTOL=0.001
```

Figure 1.Portion of the example program under test.

```
      Error on line 68 of t.f: syntax error

      Error on line 70 of t.f:no digit in statement number field

      Error on line 69 of t.f: syntax error
```

Figure 2.Example Error Messages Reported by the SGI F77 Compiler

```
      >    PIKS+=200.>

      >            ^

      t.f:INGRID line 68:

      SYNTAX ERROR #63- EXPECTED "=".
```

```
>VDGS=500;
>                ^
SYNTAX ERROR #347-semicolon in-line comments are not supported.

>fdsfRTOL=0.0001
>^
t.f:INGRID line 70:
SYNTAX ERROR #2- illegal character in label field.
```

Figure 3.Examples of Error Reported by the FORTRAN-lint

### 2.3.1 Syntax Check

Among these three software packages tested, Flint reported three syntax problems from Eagle-Surf. All of these problems are warnings for an unused label.

SYNTAX FYI #138- unused labels: 200, 901, 900

### 2.3.2 Interface Errors

Some of the interface problems reported by Flint are:

INTERFACE ERROR #55- R*4 actual arg passed to a CHAR*(*) dummy arg.

INTERFACE ERROR #130- missing subroutine: IXCLOS, IXDLSG, IXIPIK IXIPRT.

INTERFACE FYI #121- common block /HYPER/ member names differ (compared to initial use in routine COSG)

INTERFACE FYI #132- unused subroutine: INTERP1, INTERP2, INTERP3
INTERFACE FYI #279 - array passed to another of smaller size.

Although the native FORTRAN compiler does not detect many of these errors, they are indicative of poor programming and should be detected and corrected to protect the integrity of the applications.

### 2.3.3 Data Usage Errors

All of the data usage problems found are caused by variable declaration and referencing problems. An example can be seen as below:

USAGE ERROR #126- local variables referenced but never set: SCALF [Line 351]

The symbol table and the reference table generated by Flint provide comprehensive information on data being referenced and the module that referenced it. It helps the developer to spot data-referencing problems easily.

### 2.3.4 Portability Errors

Flint portability checking feature detects extensions to the FORTRAN-77 standard that may not be supported by all compilers and thus makes it easy to develop applications in an environment with many operating systems. Flint uses the FORTRAN-77 ANSI X3.9-1978 specifications. This specification is, in some cases, incompatible with the earlier X3.9-1966 standard. This is a limitation of Flint.

All three of the programs under test reported some portability errors. All errors found in Eagle-Surf and Eagle-grid were of the following types:

PORT ERROR #219- symbol names longer than 6 characters not supported by ANSI-F77.
PORT ERROR #397- ANSI-F77 does not support length specifiers on non-character data types.

Unfortunately, Flint does not provide a function to convert these variables in conflict to ones that meet the standard.

Errors reported in Ingrid were:

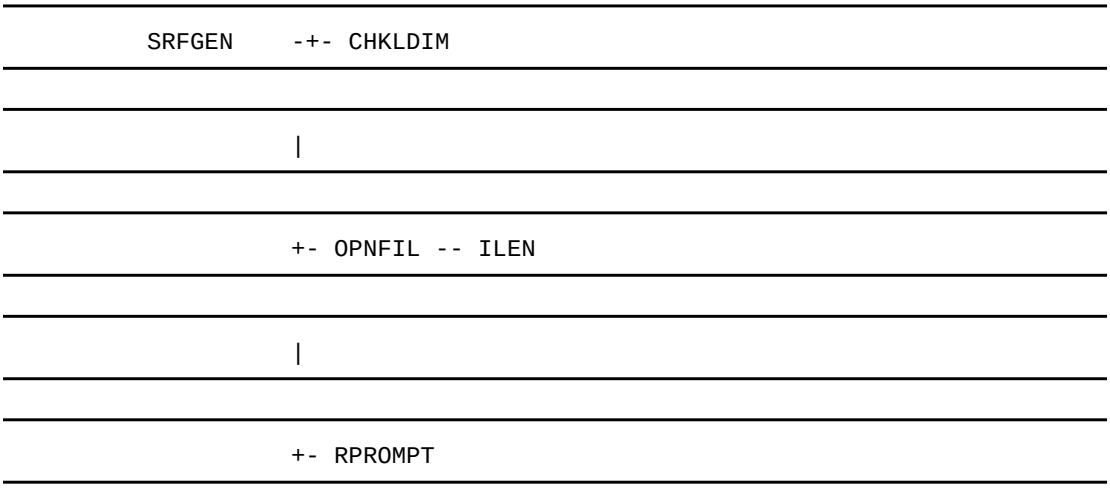
PORT ERROR #383- INCLUDE statement not supported by ANSI-F77

The SGI 4D version of Ingrid was programmed with SGI's "include" extension that allows C type include statements. Although local extensions offer some flexibility in software development under certain circumstances, include statements cause portability problems. Because software is often ported across different UNIX platforms in the NAS environment portability becomes a concern. Besides, a good programming practice is to develop software which is portable and conforms to a programming language standards. With Flint, these portability problems can easily be spotted.

### 2.3.5 Call Tree

Flint builds the call trees for Eagle-Surf, Eagle-Grid and Ingrid properly. The call trees are displayed in a structure that is easy to understand and helps users follow the calling structure without tracing the program execution. An example is shown in Figure 4.

Most of the interface errors found resulted from Ingrid, because the SGI version of Ingrid is a mixture of C programs and FORTRAN programs. Since Flint is only capable of analyzing FORTRAN programs, it does not allow analysis on FORTRAN programs that interface with C modules. In the NAS environment, a lot of software packages are developed as a mixture of FORTRAN and other languages. This is another limitation of Flint.



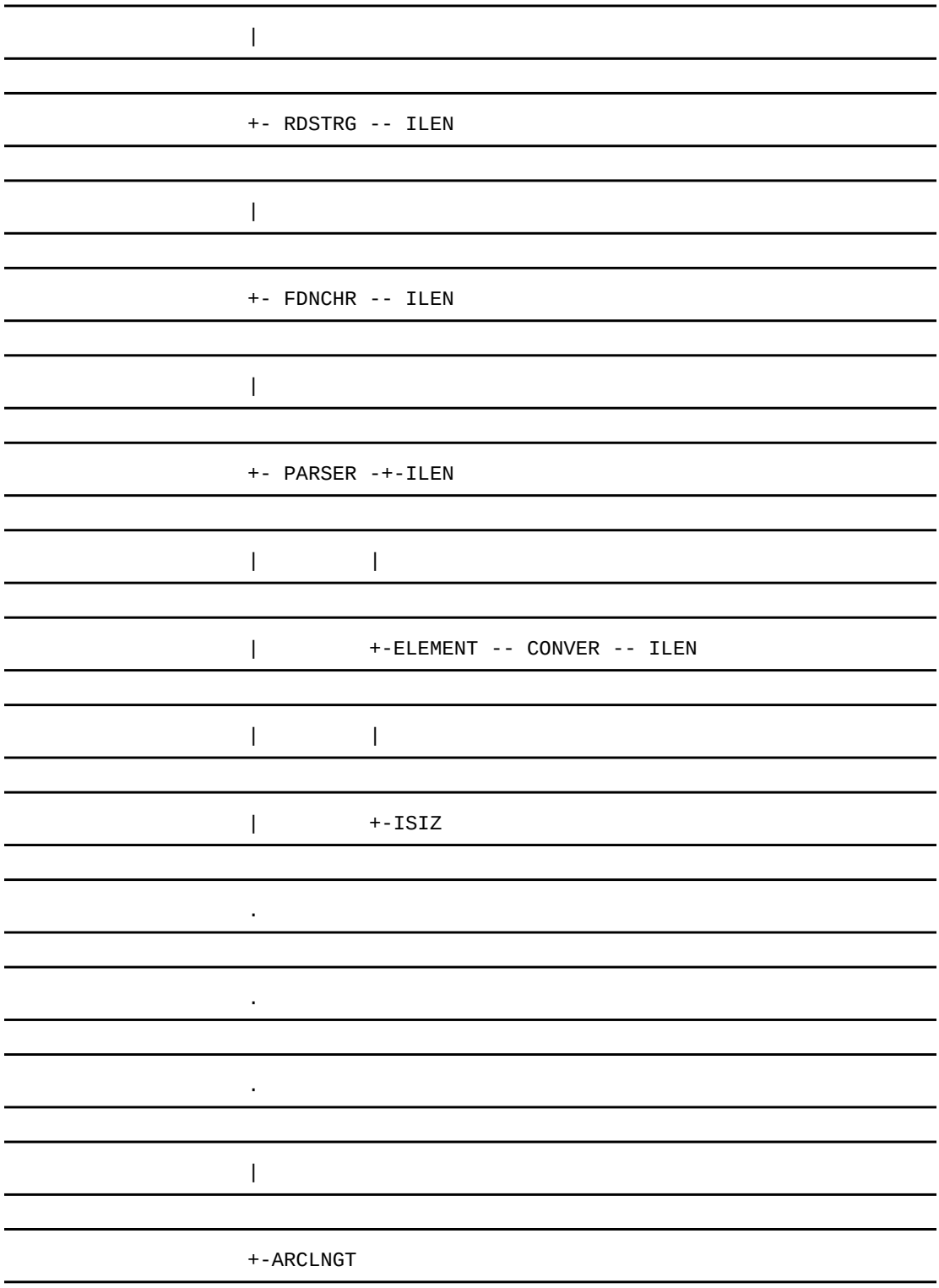



Figure 4.A Portion of the Eagle-Surf Call Tree.

## 4.0 Discussion

Flint's evaluation is completed and its contributions to NAS can be summarized as follows:

- FORTRAN-lint is helpful in detecting portability problems, syntax errors, and inconsistencies in the use of variables and subprograms. Without Flint, these problems are often undetected until much later in the development process.
- Because FORTRAN-lint helps eliminate many errors and problems before users enter the traditional debugging cycles, it helps to reduce debugging time.
- Although FORTRAN-lint does not remove errors it detects, it provides well-organized information to help users develop and maintain FORTRAN applications.

Russell Carter in the AAG group was involved in this evaluation. He is mostly interested in the portability checker of Flint. His comment on Flint is summarized as:

> Flint would be useful to RND primarily as a portability checker. The Numerical Algorithm Group (NAG[2]) FORTRAN Tools Descriptions provides portability checking as well, but also provides reformatting and transformation options, such as single-double precision converters and a program to transform a program containing long names to a program with standard names. I would recommend acquisition, or at least an on-site evaluation, of the NAG FORTRAN Tools.

The Numerical Algorithm Group (NAG) has been contacted for more information on the NAG FORTRAN Tools. It was found that these FORTRAN Tools only operate on platforms listed in table 2. There is not an implementation of this software currently available for the SGI workstations and there is no plan of porting these tools to the SGI workstations. Since most of the NAS scientific developments are performed on the SGI workstations and the Cray computers, it is a problem.

| COMPUTER SYSTEM | OPERATING SYSTEMS | CLASS |
|---|---|---|

2. NAG is a registered trademark of Numerical Algorithms Groups, Inc.

| | | |
|---|---|---|
| Alliant FX | Unix | Super computer |
| Celerity 1200 | Unix | workstation |
| Convex C1 | Unix | super computer |
| Cydrome Cydra | Unix | super computer |
| DEC MicroVax II | Unix, VMS, Ultrix | workstation |
| DEC VAX 11, 8000 | Unix, VMS, Ultrix | time sharing system |
| IBM RT PC | AIX, ACIS | workstation |
| Masscomp | Unix | workstation |
| Mutiflow | Unix | time sharing system |
| SUN | Unix | workstation |

Table 2.NAG FORTRAN Tools Environment Table

## 5.0 License Requirements

IPT FORTRAN-lint's licensing scheme is based on the number of CPUs per Auditor File which resides on a CPU or a file server. This auditor file restricts the CPUs which may access its utilities and the total number of concurrent Flint usage. For instance, a 4-user and 200-CPU auditor file allows 4 concurrent users among the 200 CPUs with permissions to access the utility. Any workstation whose name is not included in this auditor file will be denied from using it. Table 3 is a January 1991 price list of Flint.

| | Number of Users (Per Auditor File) | | | | |
|---|---|---|---|---|---|
| Number of CPUs | 1 user | 4 users | 8 users | 16 users | 32 users |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3,900 | 7,900 | 9,900 | 14,900 | 19,900 |
| up to 5 | 4,900 | | | | |
| up to 8 | | 8,900 | | | |
| up to 12 | 5,900 | | | | |
| up to 16 | | 9,900 | 11,900 | | |
| up to 25 | 6,900 | 10,900 | 12,900 | 17,900 | |
| up to 50 | 8,900 | 12,900 | 14,900 | 19,900 | |
| up to 100 | 9,900 | 13,900 | 15,900 | 20,900 | 25,900 |
| up to 200 | 11,900 | 14,900 | 16,900 | 21,900 | 26,900 |
| up to 300 | 14,900 | 18,900 | 20,900 | 25,900 | 30,900 |

Table 3.January 1991 FORTRAN-lint Price Table

Three Flint configurations are considered:

- The first approach is to install Flint on a single workstation, perhaps an SPS. Users transfer their data to the Flint server and perform the analysis on the server. Although this approach keeps a low licensing cost, it requires installation of large amount of user accounts on the Flint server, and creates extra work loads on the network and the Flint server. If a user needs to analyze a 30 MByte soft-

ware package, a total of 60 MByte round trip traffic will be created. These undesirable effects make this approach not feasible. The cost for a 4-user and 200-CPU license is $14,900; while that for an 8-user and 200-CPU license is $16,900.

- The second approach is to install the Flint on the /usr/local/bin path of the six NAS file servers. Users can retrieve and execute the Flint on their workstations. This minimizes the network traffic and the work loads on the file servers. This is the most practical but most expensive approach because six file server licenses are required. The total cost is $42,660. The advantage of this alternative is that users do not need to know the path of the Flint software; this software can be executed in an transparent manner as long as the Flint is installed on the workstation's /usr/local/bin path.

- After discussions with Michael Fredrick of IPT, a third alternative was worked out. An SGI-Cray license package, consisting of six 4-user SGI file server licenses and a 16-user Cray YMP computer license, can be purchased from IPT. This package is considered as a First-Level site license. The number of workstations which may access the Flint on these computers is unlimited; this allows any user in the NAS community to use this software, provided a user possesses a valid account on the Cray computer or a SGI workstation. The only restriction on this license package is the number of concurrent users. It allows 4 concurrent users to access Flint on the SGI file servers and 16 concurrent users on the Cray YMP computer. This translates to 24 concurrent SGI users and 16 concurrent Cray users; a total of 40 concurrent Flint users. The license cost for the SGI file servers are $47,400 and that for the Cray computer is $14,900. The total cost for this package is $62,300. IPT agrees that when the total license cost exceeds $60,000, a 50% discount will be offered to NAS and the July 1990 price schedule will be used. Based on a 50% discount rate and the cheaper July 1990 schedule, the total cost for a SGI-Cray licenses package is $31,150. This brings a substantial saving of $31,150 to NAS, and all users at NAS are allowed to use this software. The effective cost per active user is $788.75.

| License Descriptions | Total Cost | Cost/active user |
|---|---|---|
| one 4-user and 200-CPU SGI file server license | $14,900.00 | $3,725.00 |
| one 8-user and 200-CPU SGI file server license | $16,900.00 | $2,112.50 |
| six 4-user and unlimited-CPU SGI file server licenses | $42,660.00 | $1,777.50 |
| SGI-Cray licenses package at a 50% discount rate (six 4-users and unlimited-CPU SGI file server licenses and one 16-user Cray YMP license). | $31,150.00 | $ 778.75 |

Table 4.FORTRAN-lint Alternatives and Cost Comparisons.

Based on the price comparisons in table 4, a SGI-Cray licenses package at a cost of $31,150 will be the most cost effective solution to NAS since all NAS users will be allowed to use this software.

A complete Flint analysis usually takes less than a minute. Of course, the amount of analysis time depends on the size of the program and the complexity of the analysis. A 4-user license should be enough to handle the SGI users demands. This is because there are only 107 SGI workstations distributed among six SGI file servers at NAS at this time. This equates to an average of 18 users per file server. This license allows the NAS workstation population to be expanded without the need of an license upgrade.

The first year of maintenance and telephone support are included with the purchase. Subsequent software support service subscriptions are 20% of the license fee, or 8% in the case of site license. Since this SGI-Cray licenses package is considered a First-level site license, the maintenance cost is $2,492 per year after the first year.

A recent conversation with the NAG Marketing Manager has indicated that NAG is marketing a new product named *NAGWare F77 Tools*. This product is currently available for the SUN Sparcstations. NAG has committed to port this package to the SGI 4D environment and it should be available by July 1, 1991. An evaluation of this tools suite on the SUN[3] Sparcstation has been completed and its report will be available in

3. SUN is a registered trademark of SUN Microsystems, Inc.

the near future.

## 6.0 Conclusion

Information Processing Techniques' FORTRAN-lint Source Code Analyzer detects syntax problems, interface problems, data usage problems, and portability problems in correspondence to its specifications. Its portability check is particularly useful in software porting. It is a promising tool for FORTRAN-77 software developers and, at the time of this evaluation, was the only source code analyzer available for the SGI platforms.

A new product named *NAGWare F77 Tools* from NAG has become available. This product is scheduled to be ported to the SGI 4D platforms by July 1, 1991. The Beta release of NAGWare F77 tools has been evaluated on the SPS and the evaluation report[4] will be available soon. Please refer to that report for the comparison of these two tool suites and the final recommendations. In brief, the advantages of FORTRAN-lint are outweighed by the advantages of NAGWare F77 Tools; therefore, it is recommended that FORTRAN-lint not be purchased.

---

4. Lam, Terance, "Numerical Algorithms Group NAGWare F77 Tools Evaluation", NASA Ames RND Report RND-91-014, June 1991.